TITLE:

サイクル上でのグラフ探索問題に対する最適なオンラインアルゴリズム

AUTHOR(S):

森本, 尚之; 宮崎, 修一; 岡部, 寿男

CITATION:

森本, 尚之 ...[et al]. サイクル上でのグラフ探索問題に対する最適なオンラインアルゴリズム. 電子情報通信学会技術研究報告 2007, 107(219): 51-57: COMP2007-39.

ISSUE DATE:

2007-09-13

URL:

http://hdl.handle.net/2433/227011

RIGHT:

© 電子情報通信学会(IEICE)

京都大学学術情報リポジトリ KURENAI 紅
Kyoto University Research Information Repository

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
COMP2007-39 (2007-09)

サイクル上でのグラフ探索問題に対する最適なオンラインアルゴリズム

森本　尚之†　　宮崎　修一††　　岡部　寿男††

† 京都大学 情報学研究科　〒606–8501 京都府京都市左京区吉田本町
†† 京都大学 学術情報メディアセンター　〒606–8501 京都府京都市左京区吉田本町
E-mail: †morimoto@net.ist.i.kyoto-u.ac.jp, ††{shuichi,okabe}@media.kyoto-u.ac.jp

あらまし　オンライングラフ探索問題における目的は，探索者が未知のグラフの全ての頂点を訪問することによりグラフ構造を調査し，最後に出発点に戻ることである．ある辺の存在ならびにその長さは，探索者がその端点を訪れるまで未知である．目的達成に要した総移動距離を探索者のコストとして定める．探索対象を平面グラフとする場合，16 競合のアルゴリズムが知られている．朝廣らは，探索対象をサイクルグラフとする場合において，1.5 競合のアルゴリズムを与えるとともに，$(1.25-\epsilon)$ 競合のアルゴリズムは存在しないことを示した（ここで ϵ は任意の正定数である）．本稿では，サイクルグラフに対する最適なオンラインアルゴリズムを与える．すなわち，$\frac{1+\sqrt{3}}{2}(\simeq 1.366)$ 競合のアルゴリズムを与えるとともに，$(\frac{1+\sqrt{3}}{2}-\epsilon)$ 競合のアルゴリズムは存在しないことを証明する（上記と同様，ϵ は任意の正定数である）．

キーワード　オンラインアルゴリズム，競合比解析，グラフ探索問題

An Optimal Online Algorithm for the Graph Exploration Problem on Cycles

Naoyuki MORIMOTO†, Shuichi MIYAZAKI††, and Yasuo OKABE††

† Graduate School of Imformatics, Kyoto University　Yoshida-honmachi, Sakyo-ku, Kyoto, 606–8501 Japan
†† Academic Center for Computing and Media Studies, Kyoto University　Yoshida-honmachi, Sakyo-ku, Kyoto, 606–8501 Japan
E-mail: †morimoto@net.ist.i.kyoto-u.ac.jp, ††{shuichi,okabe}@media.kyoto-u.ac.jp

Abstract　The purpose of the online graph exploration problem is to visit all the nodes of a given graph and come back to the starting node with the minimum total traverse cost. However, unlike the classical traveling salesperson problem, information of the graph is given online. When an online algorithm (called a searcher) visits a node v, then it learns information on nodes and edges adjacent to v. It is known that there is a 16-competitive online algorithm for planer graphs. Recently, Asahiro et al. considered this problem on cycles and proved that there is a 1.5-competitive online algorithm, while no online algorithm can be $(1.25-\epsilon)$-competitive for any positive constant ϵ. In this paper, we give an optimal online algorithm for this problem; namely, we give a $\frac{1+\sqrt{3}}{2}(\simeq 1.366)$-competitive algorithm, and prove that there is no $(\frac{1+\sqrt{3}}{2}-\epsilon)$-competitive algorithm for any positive constant ϵ.

Key words　Online algorithm, Competitive analysis, Graph exploration problem

1. はじめに

巡回セールスマン問題 (Travelling Salesman Problem, TSP) [12] における目標は，与えられたグラフ中の全ての頂点を訪問したのちに出発点に帰ってくる経路のうちで，コスト最小のものを見つけることである．この際のコストは，経路中に含まれる辺に付与された重みの総和で定義する．TSP が NP 困難であることは広く知られており，ヒューリスティクスや近似アルゴリズムなどに対する研究が盛んに行われている．TSP の応用としては，配送業者が集荷ならびに配達を行う際の経路決定や，LSI の配線に用いるロボットのアームの動作コストの最小化といった実用的な問題が挙げられる．

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

TSP においては，グラフに関する全ての情報が巡回開始時点ですでに巡回者に与えられている．しかし，実世界の応用のなかには，アルゴリズムが実際にある場所を訪れるまでその領域に関する情報は未知であり，訪れた場所付近の局地的な情報が逐次的に与えられる，というモデルが妥当と考えられるものがある．こうした問題は**探索問題**や**地図作成問題**として知られており，[11] において無向枝重み付きグラフ上でのオンライン問題として次のように定式化されている．初期状態では探索者は出発点 o におり，o に隣接する頂点，および o と各隣接頂点とを結ぶ辺の重みのみを知識として持っている．ある頂点 v を訪れたとき，探索者は v に隣接する頂点，およびそれらと v とを結ぶ辺の重みを知る．v から u への移動は辺 $\ell(u,v)$ の重み分のコストを伴う．探索者の目的は，全ての頂点を訪問して出発点へと戻る経路のうちでコスト最小のものを発見することである．アルゴリズムの評価には**競合比解析** [5], [9] を用いる．

本問題に対する自然なアルゴリズムとしては，最近傍アルゴリズム (*Nearest Nighbour algorithm*, NN) が考えられる．これは，探索者の現在位置から最も近い距離にある未訪問頂点を次の訪問先として選択するという戦略である．しかし，頂点数 n の平面グラフに対して NN の競合比は $\Omega(\log n)$ まで大きくなることが証明されている [13]．Kalyanasundaram と Pruhs は NN を拡張したアルゴリズム ShortCut を提案し，平面グラフに対する競合比 16 を得ている [11]．

探索対象を木に限定した場合，深さ優先アルゴリズムが最適解を導くことは容易に分かる．従って，最適なオンラインアルゴリズムが自明ではない探索対象のうち，最も構造が単純なのはサイクルグラフであると考えられる．[2] においては，サイクルグラフに対する NN の競合比が 1.5 であることと，1.25 より小さな競合比を持つ決定性オンラインアルゴリズムは存在しないことが証明されている．

1.1 本研究の結果

本稿では，サイクル上でのグラフ探索問題に対するオンラインアルゴリズムの競合比の上下限を改良し，厳密な競合比 $\frac{1+\sqrt{3}}{2}(\simeq 1.366)$ を与える．上限の改良のために DIST なるアルゴリズムを示す．DIST は NN と同様に探索者が現在位置する頂点と未訪問頂点との距離を考慮に入れるが，それに加えて，探索者が探索を開始してからの総移動距離ならびに探索者が現在位置する頂点と出発点との距離を考慮して次の訪問頂点を決定する．これにより本問題に対する最適なオンラインアルゴリズムを構成することができる．

1.2 関連研究

未知の環境のオンライン探索問題は過去に広く研究されている．Deng と Papadimitriou [6]，Albers と Henzinger [1]，Fleischer と Trippen [8] は未知の無向グラフ上での探索問題を考えた．ただし，それらの問題における探索者の目的は全ての頂点を訪れることではなく全ての辺を渡ることであり，探索者のコストは渡った辺の本数と定義されている．[7], [10] では探索対象を多角形の内部としている．探索者の目的は，多角形が持つ全ての境界辺を発見したのち出発点へと戻ることである．[3], [4] では，訪問するべき頂点の要求がオンライン的に与えられるという条件を加えた場合の TSP について研究されている．ここでの探索者の目的は，要求された各頂点を訪れることである（ただし，k サーバ問題とは異なり，要求が与えられた順番の通りに訪問する必要は無い）．

2. 準　　備

オンライングラフ探索問題における目的は，グラフ $G=(V,E)$（V は頂点集合，E は辺集合を表す）中の全ての頂点を訪問することである．各枝 $(u,v)\in E$ には非負の重み $\ell(u,v)$ が付与されている．以下ではこの重みを長さと呼ぶ．初期状態では探索者は出発点 $o\in V$ に位置する．この時点で探索者が持つ情報は，o に隣接する 2 つの頂点ならびにそれら頂点と o とを結ぶ辺の長さである．探索者は，ある頂点 v を訪れたときに v に隣接する 2 つの頂点ならびにそれら頂点と v とを結ぶ辺の長さを知る．また，探索者の持つ記憶容量は十分大きく，探索開始後に得たグラフに関する全ての情報を保持することができる．探索者は，各時点において得ている知識のみを用いて次にどの頂点を訪問するかを決定しなければならない．探索者の目標は，グラフ上の全ての頂点を訪問した後に出発点へと戻ることである．その達成に要した総移動距離を探索者のコストとする．

オンラインアルゴリズムの性能評価には**競合比解析**を用いる．あるアルゴリズム ALG によって見つけられたグラフ G 上の経路の長さを ALG(G)，最適なオフラインアルゴリズムが出力する G 上の経路の長さを OPT(G) とする．任意の $G\in\mathcal{G}$ について，ALG(G)/OPT(G) $\leqq c$ が成立するならば，グラフクラス $\mathcal{G}$ に対して ALG は c 競合であると言い，c を競合比と呼ぶ．以下，混乱の恐れが特に無い場合，ALG(G)，OPT(G) をそれぞれ単に ALG，OPT と書く．

本問題における，単純ながら重要な事実が [2] にて示されている．ℓ_{max} を G 中の最長辺とし，$L=\sum_{(u,v)\in E}\ell(u,v)$ を全ての辺の長さの総和とする．

[事実 1]　任意のサイクルグラフ C において，$\ell_{max}\leq\frac{L}{2}$ ならば OPT$(C)=L$ であり，$\ell_{max}>\frac{L}{2}$ ならば OPT$(C)=2(L-\ell_{max})$ である．

3. 決定性アルゴリズムに対する競合比の下限

本節では，本問題に対する任意のオンラインアルゴリズムの競合比の下限を与える．

[定理 1]　任意の正定数 ϵ に対し，$\frac{1+\sqrt{3}}{2}-\epsilon$ 競合であるオンラインアルゴリズムは存在しない．

証明．オンラインアルゴリズムの挙動に対して，アドバーサリーとしてグラフを提示することで上記の下限を導く．n は $n>\frac{\sqrt{3}}{2\epsilon}$ を満たすある整数とする．初期状態においてアドバーサリーは，長さ 1 を持つ 2 つの枝 (o,u_1) と (o,v) を探索者に提示する．対称性より探索者は u_1 に移動するとして一般性を失わない．次いでアドバーサリーは $\ell(u_1,u_2)=1$ を満たす辺 (u_1,u_2) を提示する．もしも探索者が u_2 を訪問した場合，$\ell(u_2,u_3)=1$ を満たす辺 (u_2,u_3) を提示する．以下同様に，探索者が新しい頂点 u_i $(i\leqq n-1)$ を訪問したときにアドバーサ

A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp

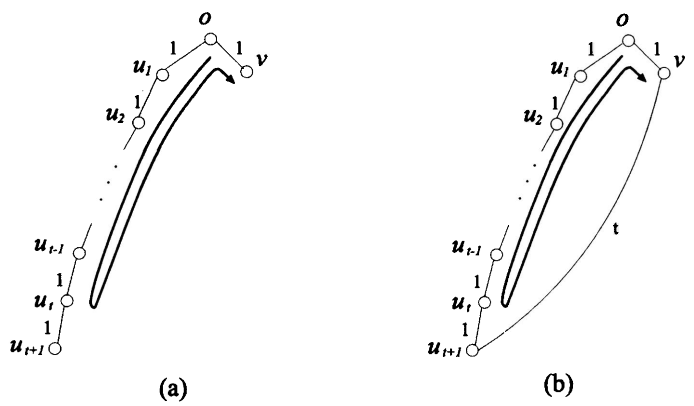

図 1　オンラインアルゴリズムに対するアドバーサリーの提示

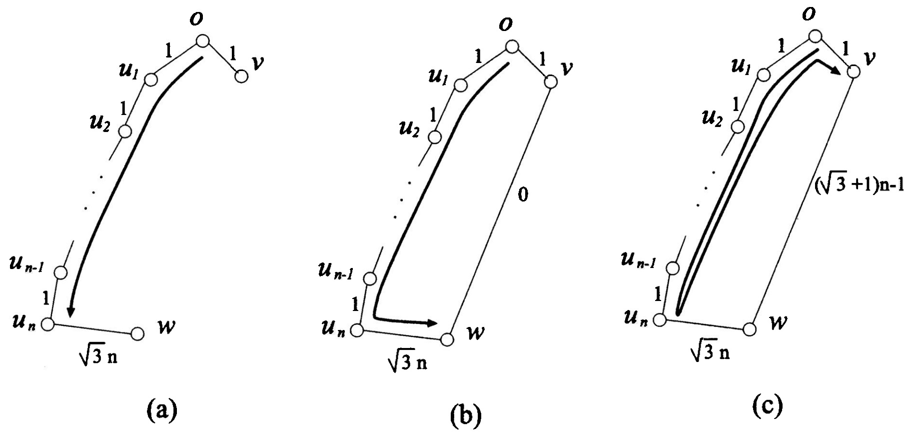

図 2　オンラインアルゴリズムに対するアドバーサリーの提示

リーは $\ell(u_i, u_{i+1})$ を満たす辺 (u_i, u_{i+1}) を提示する．

まず，探索者が u_n を訪れる前に v を訪問した場合を考える．この場合，探索者はある頂点 u_t $(t \leqq n-1)$ を訪れた直後に（すなわち，辺 (u_t, u_{t+1}) を見たのち，引き返して）v を訪問している (図 1 (a))．これに対しアドバーサリーは $\ell(v, u_{t+1}) = t$ なる辺 (v, u_{t+1}) を提示する (図 1 (b))．この後，唯一の未訪問頂点である u_{t+1} を訪問して出発点に戻る経路のうちで探索者にとって最適なのは，v から辺 (v, u_{t+1}) を渡って u_{t+1} を訪れ，時計回りまたは反時計回りの道をたどって出発点に戻る，という経路である．このときの探索者の総コストは $4t+2$ である．一方，最適なオフラインアルゴリズムは単純にサイクルを一周するので，そのコストは $2t+2$ である．これらより，競合比は $\frac{4t+2}{2t+2} \geqq 1.5$　($t \geqq 1$ より) である．

次に，探索者が v を訪問する前に u_n に到達する場合を考える．このときアドバーサリーは $\ell(u_n, w) = \sqrt{3}n$ を満たす辺 (u_n, w) を提示する (図 2 (a))．ここで，探索者が次に訪問する頂点は w または v である．

w を訪問する場合，アドバーサリーは $\ell(w, v) = 0$ を満たす辺 (w, v) を提示する (図 2 (b))．この後 (w, v), (v, o) の順番で辺を渡って出発点に帰るのが探索者にとって最適な経路であり，総コストは $n+\sqrt{3}n+0+1 = (1+\sqrt{3})n+1$ である．$\ell(u_n, w)$ はサイクルグラフの全長の半分よりも大きいので，事実 2.1 より最適なオフラインアルゴリズムが出力する経路のコストは $2(n+1)$ である．よって競合比は，$\frac{(1+\sqrt{3})n+1}{2(n+1)} = \frac{1+\sqrt{3}}{2} - \frac{\sqrt{3}}{2(n+1)} > \frac{1+\sqrt{3}}{2} - \epsilon$ である．

一方，辺 (u_n, w) が提示された時点で探索者が引き返して v を訪問した場合は，アドバーサリーは $\ell(v, w) = (\sqrt{3}+1)n - 1$ を満たす辺 (v, w) を提示する (図 2 (c))．最後の未訪問頂点は w であるから，(v, w) を渡って w を訪問し，その後時計回りまたは反時計回りの道を通って w から o に戻るのが探索者にとって最適な経路である．よって，経路の総コストは $n+n+1+(\sqrt{3}+1)n-1+(\sqrt{3}+1)n = (2\sqrt{3}+4)n$ である．最適なオフラインアルゴリズムは単純にサイクルを一周するので，そのコストは $(2\sqrt{3}+2)n$ である．よって，競合比は $\frac{(2\sqrt{3}+4)n}{(2\sqrt{3}+2)n} = \frac{1+\sqrt{3}}{2}$ である．　□

4. アルゴリズム DIST とその競合比の上限

本節ではオンラインアルゴリズム DIST を定義し，その競合比が $\frac{1+\sqrt{3}}{2}$ であることを示す．

4.1　アルゴリズム DIST

探索対象のグラフはサイクルであるから，各時点において探索者が次に訪れる頂点の選択肢は常に 2 つ存在する (図 3 (a) 参照．ここで，点線で囲まれた部分は訪問済みの領域であり，黒く塗りつぶされた頂点は探索者の現在位置を示している．これ以下の図も同様)．

アルゴリズム DIST を示す前にいくつかの記号を定義する．

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

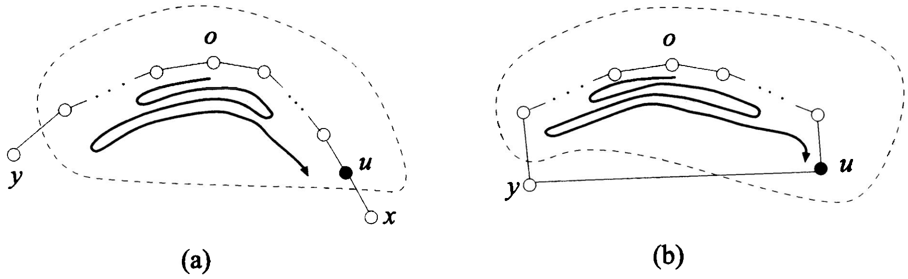

図 3　アルゴリズム DIST

図 3 (a) において探索者は現在 u に位置しており，x と y のうちどちらを次に訪問するかを決定する．ある 2 つの頂点 v_1 と v_2 とを結ぶ，既知の辺のみによって構成される道の長さを $d(v_1, v_2)$ とする．また，探索者が探索を開始してから移動した総距離を X とする．ここで $W = X - d(o,u)$ とおく．W の値は探索が進むにつれて変化するので，アルゴリズムのステップ i における W の値は例えば W_i とでも表記すべきものである．しかし簡単のため，混乱の恐れが無い場合，単に W と表記するか，もしくは「その時点での W 値」と呼ぶことにする．

ステップ 1:　探索者は出発点 o に位置し，o には 2 つの辺が隣接している．探索者は o に近い方の頂点を訪れる．2 つの頂点への距離が等しい場合，任意の 1 つを選ぶ．

ステップ i ($i \geqq 2$):　図 3 (a) において探索者が u に位置しているとする．もし $\ell(u,x) \leqq \sqrt{3}d(u,y) - W$ であれば，探索者は x に移動する．$\ell(u,x) > \sqrt{3}d(u,y) - W$ ならば，y に移動する．

最終ステップ:　状況を図 3 (b) に示す．探索者は u を訪れた際，u は既知の未訪問頂点 y に隣接していることを知る．この時点でグラフの全体が判明し，最後の未訪問頂点は y である．u から y への移動ならびに y から o への移動は，それぞれ距離の短い道を選択して行う．

4.2　競合比解析

本節では次の定理を証明する．

[定理 2]　DIST の競合比は高々 $\frac{1+\sqrt{3}}{2}$ である．

DIST のあるステップにおける図 3 (a) のような状況を考える．ただし，最終ステップの直前においては図 3 (b) に示すように x と y とは同じものである．次に示す補題は以下の証明において重要である．

[補題 1]　$W \leqq (\sqrt{3}-1)d(o,y)$.

注意: 図 3 (b) においては，$d(o,u)$ は o から u までの時計回りに計った距離であり，$d(o,y)$ は o から y までの反時計回りに計った距離を表すものとする．

証明．証明は帰納法を用いる．第 1 ステップの後の状況を図 4 (a) に示す．$W = \ell(o,u) - \ell(o,u) = 0$ であるから与式は成立する．

次にステップ i で与式が成立すると仮定し，ステップ $i+1$ での成立を示す．ステップ i の後の状況を図 3 (a) に示す．この時点での W 値を W_i とする．帰納法の仮定により $W_i \leqq (\sqrt{3}-1)d(o,y)$ が成り立つ．このとき探索者が次に x と y のどちらを訪れるかによって 2 つのケースが考えられる．

Case 1. ステップ $i+1$ において探索者が x に移動したときの状況を図 4 (b) に示す．W 値の定義により，現在の W 値すなわち W_{i+1} は，$W_{i+1} = W_i + \ell(u,x) - \ell(u,x) = W_i$ と求まる．これは，探索者が探索方向を変えない限り W 値は変化しないことを意味している（この性質は後にも用いる）．従って $W_{i+1} \leqq (\sqrt{3}-1)d(o,y)$ であり，ステップ $i+1$ でも与式は成立している．

Case 2. ステップ $i+1$ において探索者が y に移動したときの状況を図 4 (c) に示す．このステップにおける探索者の移動距離は $d(y,o) + d(o,u)$ である．また，原点からの距離は $d(o,u)$ であったが，この移動により $d(o,y)$ へと変化している．従って $W_{i+1} = W_i + d(y,o) + d(o,u) - d(o,y) + d(o,u) = W_i + 2d(o,u)$．探索者は移動先として x ではなく y を選んだので $\ell(u,x) > \sqrt{3}d(u,y) - W_i$ が成り立つ．また，帰納法の仮定により $W_i \leqq (\sqrt{3}-1)d(o,y)$．これら 2 式と $d(u,y) = d(u,o) + d(o,y)$ より $d(o,y) < \ell(u,x) - \sqrt{3}d(u,o)$ を得る．再び帰納法の仮定を用いて

$$
\begin{aligned}
W_{i+1} &= W_i + 2d(o,u) \\
&\leqq (\sqrt{3}-1)d(o,y) + 2d(o,u) \\
&< (\sqrt{3}-1)(\ell(u,x) - \sqrt{3}d(u,o)) + 2d(o,u) \\
&= (\sqrt{3}-1)(\ell(u,x) + d(o,u)) \\
&= (\sqrt{3}-1)d(o,x)
\end{aligned}
$$

となる．よってステップ $i+1$ でも与式は成立している．　□

さて，最終ステップの直前の状態を考える．この状況を図 5 に示す．探索者は u に位置し，この時点で，u に隣接する頂点 y は以前頂点 v から見たものと同じであることを知る．$\ell(u,y)$ を知ったので，探索者はグラフに関する全情報を得たことになる．

簡単のため，図 5 に示すように $d(o,u)$，$\ell(u,y)$，$d(o,v)$，$\ell(v,y)$ の長さをそれぞれ a，b，c，d と表す．この時点での W 値を W^* とする．補題 1 より $W^* \leqq (\sqrt{3}-1)d(o,y) = (\sqrt{3}-1)(c+d)$ が成り立つ．唯一の未訪問頂点は y であり，それを訪問する際は y への道のうち距離の短い方を選ぶ．最後に，時計回りの道と反時計回りの道のうち距離の短い方を通って y から o へと移動する．以下，場合分けを行う．

Case 1. $b > a + c + d$ の場合．

このとき，$a+b > c+d$ が成り立つ．従って探索者は反時計回りの道を通って y を訪問し，最後に時計回りの道を通って o に戻る．最終ステップの直前の時点までに探索者が移動した距離の総

A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp

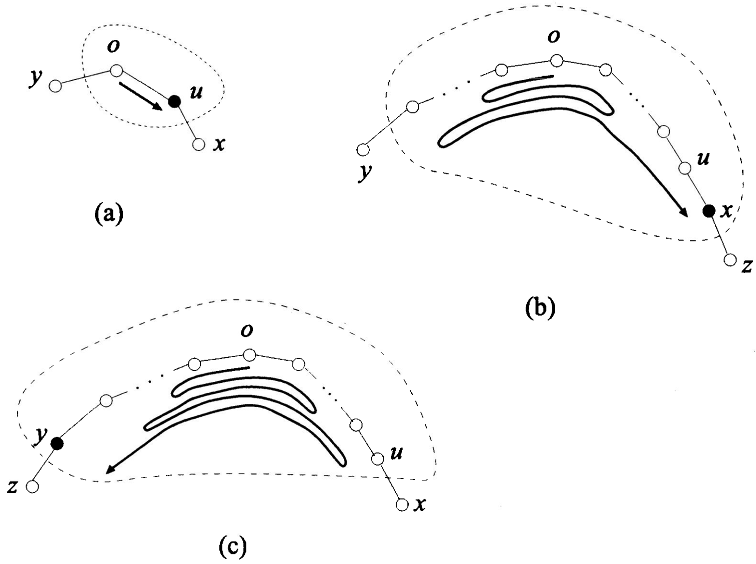

図 4　補題 1 の証明

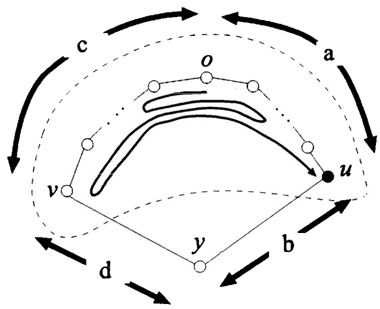

図 5　DIST の最終ステップ直前における状態

和は，W 値の定義により $W^*+d(o,u)=W^*+a$ である．従って $\mathsf{DIST}=W^*+a+(a+c+d)+(c+d)=W^*+2(a+c+d)$．$b>a+c+d$ より $\ell_{max}=b>L/2$ であるから，事実 1 より $\mathsf{OPT}=2(a+c+d)$．従って競合比は

$$\begin{aligned}\frac{\mathsf{DIST}}{\mathsf{OPT}} &= \frac{W^*+2(a+c+d)}{2(a+c+d)}\\ &\leqq 1+\frac{(\sqrt{3}-1)(c+d)}{2(a+c+d)}\\ &\leqq 1+\frac{\sqrt{3}-1}{2}\\ &= \frac{1+\sqrt{3}}{2}.\end{aligned}$$

Case 2. $b \leqq a+c+d$ かつ $a+b \leqq c+d$ の場合．

探索者は辺 (u,y) を通って y を訪問し，その後反時計回りの道を通って o に戻る．従って $\mathsf{DIST}=W^*+a+b+(b+a)=W^*+2(a+b)$ である．ここでグラフ中の最長辺を e_{max} とする（すなわち $\ell(e_{max})=\ell_{max}$）．以下では，e_{max} の長さならびにその存在場所に応じてさらに場合を分けて考える．

Case 2-(i). $\ell_{max} \leqq L/2$ の場合．

事実 1 より $\mathsf{OPT}=a+b+c+d$．従って

$$\begin{aligned}\frac{\mathsf{DIST}}{\mathsf{OPT}} &= \frac{W^*+2(a+b)}{a+b+c+d}\\ &\leqq \frac{(\sqrt{3}-1)(c+d)+2(a+b)}{a+b+c+d}\\ &= \sqrt{3}-1+\frac{(3-\sqrt{3})(a+b)}{a+b+c+d}\\ &\leqq \sqrt{3}-1+\frac{(3-\sqrt{3})(a+b)}{2(a+b)}\\ &= \frac{1+\sqrt{3}}{2}.\end{aligned}$$

Case 2-(ii). $\ell_{max}>L/2$ かつ $e_{max}=(u,y)$ の場合．

$b \leqq a+c+d$ より，この場合は起こりえない．

Case 2-(iii). $\ell_{max}>L/2$ かつ $e_{max}=(v,y)$ の場合．

事実 1 より $\mathsf{OPT}=2(a+b+c)$．ここで，探索者が v にいたときの状況を考える（図 6 (a)）．このときの W 値を W' とする．補題 1 より $W' \leqq (\sqrt{3}-1)d(o,u'') \leqq (\sqrt{3}-1)a$ である．y が最後の未訪問頂点であるから，次のステップで探索者は u'' に移動する．探索者が u'' にいるときの W 値を W'' とする．この移動によって生じる総移動距離の増加分は $c+d(o,u'')$ であり，探索者の現在位置と出発点との距離は c から $d(o,u'')$ へと変化する．したがって $W''=W'+c+d(o,u'')+c-d(o,u'')=W'+2c$．この後 u に到達するまで探索者は進行の方向を変えないので，u に到達するまで W 値は不変であり，$W^*=W''=W'+2c \leqq (\sqrt{3}-1)a+2c$．従って

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

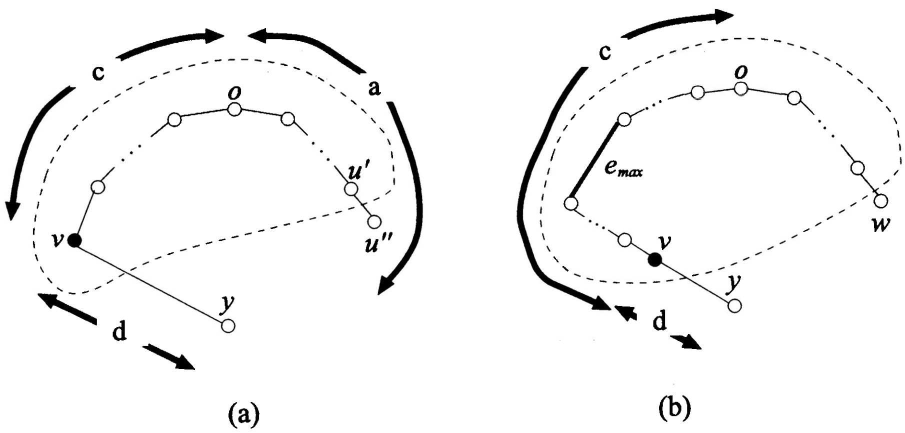

図 6　Case 2-(iii) と Case 2-(iv)

$$
\begin{aligned}
\frac{\mathsf{DIST}}{\mathsf{OPT}} &= \frac{W^* + 2(a+b)}{2(a+b+c)} \\
&\leqq \frac{(\sqrt{3}-1)a + 2c + 2(a+b)}{2(a+b+c)} \\
&= 1 + \frac{(\sqrt{3}-1)a}{2(a+b+c)} \\
&\leqq 1 + \frac{\sqrt{3}-1}{2} \\
&= \frac{1+\sqrt{3}}{2}.
\end{aligned}
$$

Case 2-(iv). $\ell_{max} > L/2$ かつ，e_{max} が o から v への反時計回りの道の中に含まれている場合．

この場合は起こりえないことを背理法で証明する．e_{max} が o から v への反時計回りの道の中に含まれていると仮定する．探索者が v にいたときの状況を考える (図 6 (b))．探索者は y でなく w に移動したので，この時点での W 値を W' とすると $\ell(v,y) > \sqrt{3}d(v,w) - W'$ が成り立っている．また，補題 1 により $W' \leqq (\sqrt{3}-1)d(o,w)$．背理法における仮定より $d(v,o) \geqq \ell_{max}$ であるから，$\ell(v,y) > \sqrt{3}d(v,w) - (\sqrt{3}-1)d(o,w) = \sqrt{3}d(v,o) + d(o,w) \geqq \ell_{max}$ となり矛盾．従ってこの場合は起こりえない．

Case 2-(v). $\ell_{max} > L/2$ かつ， e_{max} が o から u への時計回りの道の中に含まれている場合．

$a+b \leqq c+d$ より，この場合は起こりえない．

Case 3. $b \leqq a+c+d$ かつ $a+b > c+d$．

探索者は辺 (u,y) を渡って y を訪れ，その後時計回りの道を辿って o に戻る．よって $\mathsf{DIST} = W^* + a + b + (c+d)$．以下，$e_{max}$ の長さならびにその存在位置に応じて場合を分けて考える．

Case 3-(i). $\ell_{max} \leqq L/2$ の場合．

事実 1 より $\mathsf{OPT} = a+b+c+d$．よって

$$
\begin{aligned}
\frac{\mathsf{DIST}}{\mathsf{OPT}} &= \frac{W^* + a+b+c+d}{a+b+c+d} \\
&\leqq 1 + \frac{(\sqrt{3}-1)(c+d)}{a+b+c+d} \\
&< 1 + \frac{(\sqrt{3}-1)(c+d)}{2(c+d)} \\
&= \frac{1+\sqrt{3}}{2}.
\end{aligned}
$$

Case 3-(ii). $\ell_{max} > L/2$ かつ $e_{max} = (u,y)$ の場合．

$b \leqq a+c+d$ より，この場合は起こりえない．

Case 3-(iii). $\ell_{max} > L/2$ and $e_{max} = (v,y)$ の場合．

$a+b > c+d$ より，この場合は起こりえない．

Case 3-(iv). $\ell_{max} > L/2$ かつ， e_{max} が o から v への反時計回りの道の中に含まれている場合．

$a+b > c+d$ より，この場合は起こりえない．

Case 3-(v). $\ell_{max} > L/2$ かつ， e_{max} が o から u への時計回りの道の中に含まれている場合．

探索者が v にいたときの状況を考える．まず，探索者はこの時点では e_{max} を渡っていないことを背理法で証明する．探索者がすでに e_{max} を渡っていると仮定する （図 7(a)）．探索者は次のステップで w を訪れたので，この時点での W 値を W' とすると $\ell(v,y) > \sqrt{3}d(v,w) - W'$．補題 1 より，$W' \leqq (\sqrt{3}-1)d(o,w)$．よって $\ell(v,y) > \sqrt{3}d(v,w) - (\sqrt{3}-1)d(o,w) = \sqrt{3}d(o,v) + d(o,w) \geqq \ell_{max}$ となるが，これは矛盾である．よって，探索者は v を離れたのちにはじめて e_{max} を渡る．

$e_{max} = (u', u'')$ として，探索者が u' にいたときの状況を考える （図 7 (b)）．この時点での W 値を W'' とする．探索者は次に u'' を訪れるから，

$$
\begin{aligned}
\ell_{max} &\leqq \sqrt{3}d(u',y) - W'' \\
&= \sqrt{3}(d(o,u') + d(o,y)) - W'' \\
&\leqq \sqrt{3}(d(o,u) - \ell_{max} + d(o,y)) - W''.
\end{aligned}
$$

最後の不等式は $d(o,u') + \ell_{max} \leqq d(o,u)$ による．従って

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

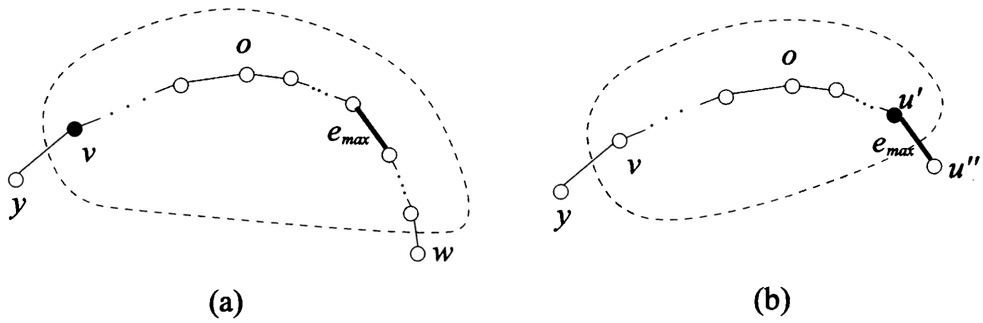

図 7　Case 3-(v)

$$
\begin{aligned}
\ell_{max} &\leqq \frac{\sqrt{3}(d(o,u)+d(o,y))-W''}{1+\sqrt{3}} \\
&= \frac{\sqrt{3}(a+c+d)-W''}{1+\sqrt{3}} \\
&= \frac{\sqrt{3}(L-b)-W''}{1+\sqrt{3}}.
\end{aligned}
$$

ここで L はサイクルの全長である．y が最後の未訪問頂点であるから，探索者は u に到達するまで進行の方向を変えていない．よって $W^* = W''$ である．一方，事実 1 より $\mathsf{OPT} = 2(L-\ell_{max})$．従って

$$
\begin{aligned}
\frac{\mathsf{DIST}}{\mathsf{OPT}} &= \frac{W^*+a+b+c+d}{2(L-\ell_{max})} \\
&\leqq \frac{W^*+L}{2(L-\frac{\sqrt{3}(L-b)-W''}{1+\sqrt{3}})} \\
&= \frac{(1+\sqrt{3})(W^*+L)}{2(L+W^*+\sqrt{3}b)} \\
&\leqq \frac{1+\sqrt{3}}{2}.
\end{aligned}
$$

5. おわりに

本稿では，サイクルグラフ上のオンライングラフ探索問題に対する厳密な競合比 $\frac{1+\sqrt{3}}{2}$ を得た．次の考察対象となるグラフとしてはカクタスが挙げられる．別の研究方向としては乱択アルゴリズムを用いることが考えられる．さらには，今回用いた方法を平面グラフに適用することで現在の上限値 16 を改良することについても研究してゆきたい．

文　献

[1] S. Albers and M.R. Henzinger, "Exploring unknown environments," *SIAM J. Computing* **29** (4), pp.1164–1188, 2000.

[2] Y. Asahiro, E. Miyano, S. Miyazaki and T. Yoshimuta, "Weighted nearest neighbor algorithms for the graph exploration problem on cycles," In *Proc. 33rd SOFSEM 2007* (LNCS 4362), pp. 164–175, 2007.

[3] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, M. Talamo, "Algorithms for the on-line traveling salesman," *Algorithmica* **29** (4), pp.560–581, 2001.

[4] G. Ausiello, V. Bonifaci and L. Laura, "The on-line asymmetric traveling salesman problem," *Proc. 9th WADS*, pp.306–317, 2005.

[5] A. Borodin and R. El-Yaniv, "Online Computation and Competitive Analysis," *Cambridge University Press*, 1998.

[6] X. Deng and C.H. Papadimitriou, "Exploring an unknown graph," In *Proc. 31st FOCS*, pp.355–361, 1990.

[7] X. Deng, T. Kameda, C.H. Papadimitriou, "How to learn an unknown environment," In *Proc. 32nd FOCS*, pp.298–303, 1991.

[8] R. Fleischer and G. Trippen, "Exploring an unknown graph efficiently," In *Proc. 13th ESA* (LNCS 3669), pp.11–22, 2005.

[9] A. Fiat and G.J. Woeginger, "Competitive analysis of algorithms," In *Online Algorithms: The State of the Art* (Fiat and Woeginger Ed., Springer), 1998.

[10] F. Hoffmann, C. Icking, R. Klein, K. Kriegel, "The polygon exploration problem," *SIAM J. Computing* **31** (2), pp.577–600, 2001.

[11] B. Kalyanasundaram and K.R. Pruhs, "Constructing competitive tours from local information," *Theoretical Computer Science* **130**, pp.125–138, 1994.

[12] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoyes (eds.), "The traveling salesman problem: a guided tour of combinatorial optimization," Wiley, Chichester, 1985.

[13] D.J. Rosenkrantz, R.E. Stearns, P.M. Lewis, "An analysis of several heuristics for the traveling salesman problem," *SIAM J. Computing* **6** (3), pp.563–581, 1977.